|  |  |
| --- | --- |
| **学 号：** | 0122010870309 |

****

**软件工程实践（二）**

|  |  |
| --- | --- |
| **学 院** | **计算机与人工智能学院** |
| **专 业** | **软件工程** |
| **班 级** | **软件2002** |
| **姓 名** | **毕列祥** |
| **编 号** | **0122010870309** |
| **指导教师** | **唐祖锴** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2023 | 年 | 6 | 月 | 18 | 日 |

**目录**

[1 任务概述 1](#_Toc91495513)

[2 任务分析 1](#_Toc91495514)

[3 开发计划 1](#_Toc91495515)

[4 软件配置计划 1](#_Toc91495516)

[5 测试计划 1](#_Toc91495517)

[6 实施情况 1](#_Toc91495518)

[7 实施过程问题记录与分析 2](#_Toc91495519)

[8 任务总结 2](#_Toc91495520)

[9 参考文献 2](#_Toc91495521)

# 1 任务概述

**任务目的**

\* 巩固强化软件编程规范

\* 提高面向对象软件建模与抽象能力

\* 培养小组协同开发能力

\* 掌握基于Maven的软件项目管理机制

\* 掌握基于Github的小组协同开发工具和平台

\* 了解DevOps软件开发流程

**任务内容**

\* 创建软件开发小组

\* 针对样例代码工程进行小组讨论，确定功能扩充需求点

\* 基于Github中的issue管理功能明确工作任务并为组员分配工作任务

\* 基于小组商定的分支模型进行软件功能开发，并按开发流程进行代码测试、提交、归并和同步

\* 代码提交到远程仓库后，应进行自动化代码格式规范检查和测试以确保功能符合需求设计

\* 完成前述各项任务后，可尝试进行代码自动化打包，自动生成可供执行的jar文件

**任务要求**

1. 创建软件开发小组

\* 每个开发小组人数3-5人，推选一人作为组长，负责组织、协调和领导团队开发；

\* 所有小组成员应按操作步骤在github开发平台上加入同一小组，共用同一代码仓库；

2. 开展小组讨论，确定功能扩充点

\* 样例工程“world-of-zuul”具备最基本的程序功能，该项目具有极大的扩展空间，开发小组内可进行沟通讨论，确定系统结构优化需求或功能扩充需求，结构优化或功能扩充项不能少于5项；

> 可供参考的结构优化或功能扩充项包括但不限于以下内容：

>

> 1. 扩展游戏，使得一个房间里可以存放任意数量的物件，每个物件可以有一个描述和一个重量值，玩家进入一个房间后，可以通过“look”命令查看当前房间的信息以及房间内的所有物品信息；

> 2. 在游戏中实现一个“back”命令，玩家输入该命令后会把玩家带回上一个房间；

> 3. 在游戏中实现一个更高级的“back”命令，重复使用它就可以逐层回退几个房间，直到把玩家带回到游戏的起点；

> 4. 在游戏中增加具有传输功能的房间，每当玩家进入这个房间，就会被随机地传输到另一个房间；

> 5. 在游戏中新建一个独立的Player类用来表示玩家，并实现下列功能需求：

> \* 一个玩家对象应该保存玩家的姓名等基本信息，也应该保存玩家当前所在的房间；

> \* 玩家可以随身携带任意数量的物件，但随身物品的总重量不能操过某个上限值；

> \* 在游戏中增加两个新的命令“take”和“drop”，使得玩家可以拾取房间内的指定物品或丢弃身上携带的某件或全部物品，当拾取新的物件时超过了玩家可携带的重量上限，系统应给出提示；

> \* 在游戏中增加一个新的命令“items”, 可以打印出当前房间内所有的物件及总重量，以及玩家随身携带的所有物件及总重量；

> \* 在某个或某些房间中随机增加一个magic cookie（魔法饼干）物件，并增加一个“eat cookie”命令，如果玩家找到并吃掉魔法饼干，就可以增长玩家的负重能力；

> 6. 扩充游戏基本架构，使其支持网络多人游戏模式，具备玩家登陆等功能；

> 7. 为单机或网络版游戏增加图形化用户界面，用过可以通过图形化界面执行游戏功能；

> 8. 可以为游戏增加数据库功能，用于保存游戏状态和用户设置；

> 9. ......

3. 基于Github中的issue管理功能明确工作任务并为组员分配工作任务

\* 将工作任务拆分细化后，明确版本开发计划和里程碑时间节点；

\* 在github平台创建任务issue并为所有组员分配任务；

\* 每位组员可以分别承担不同的开发任务，也可以按照小组角色分别承担开发、测试、集成等工作任务；

\* 工作任务的划分是最终衡量小组成员工作量的重要依据；

4. 基于小组商定的分支模型进行软件功能开发，并按开发流程进行代码测试、提交、归并和同步

\* 小组成员按照小组商定的分支模型在各自的工作分支进行进行开发任务；

\* 工作分支在合并前应同步到远程仓库供教师检查每人的开发工作完成情况；

\* 提交代码时应按照小组约定的规范格式填写代码提交说明，代码提交说明也将作为评分的重要依据；

5. 代码提交到远程仓库后，应进行自动化代码格式规范检查和测试以确保功能符合需求设计；

\* 可以利用github平台的actions功能在代码提交时自动触发代码格式检查，对于不符合规范的代码系统将给出提交失败提示；

\* 可以利用github平台的actions功能在代码提交时自动触发测试用例检查，对于不能通过测试检查的代码系统将给出提交失败提示；

6. 可尝试进行代码自动化打包，自动生成可供执行的jar文件

\* 结合github平台的actions功能和maven编译脚本，在代码通过规范性检查和测试用例后，进行自动化打包，生成可供直接执行的jar文件用于系统发布

# 2 任务分析

本次实践任务我分配到的任务是前端登录界面的开发



本次实践任务的技术方案:

利用VSCODE开发工具来实施整个项目进程；面向对象程序设计方法在实践工程项目中的应用；利用Git工具支持个人代码版本管理；利用CodeStyle等代码规范检查工具进行规范性约束和检查；基于Junit框架实施单元测试和测试驱动开发。

本次实践任务的工具使用方案:

版本控制工具：git

编写工具:vscode

测试工具：Junit

管理工具：maven，vue

本次实践任务的重难点分析:

项目中期对代码的合并处理比较困难，每个人编写的代码单独运行都不成问题，合并后各类调用处理起来很麻烦;加上对于github网站操作不熟练，合并代码时候产生很多冲突；添加拓展功能并优化的同时需要一些经验，不然根本无从下手。

# 3 开发计划

项目开发计划：

1.需求分析：首先需要明确用户需求，包括游戏界面设计、操作流程、游戏功能等。通过用户调研和竞品分析，确定用户喜好和市场特点。

2.界面设计：设计游戏界面的UI，包括菜单栏、游戏画面、字体颜色、位图等。将游戏画面和菜单设置为可定制和可调整的，方便用户个性化设置。

3.技术选型：选择一种前端框架和其他工具，例如 React、Vue.js 或 Angular 等。确保框架能够快速地处理不同场景下的页面渲染、用户输入、事件处理等模块需求。

4.架构设计：设计前端架构，遵循渐进增强、可访问性、组件化、可测试性等原则。同时，应当考虑如何设计一个可扩展的系统，从而使游戏更易于维护、更新和升级。

5.编码实现：基于设计确定的UI，采用所选的前端框架和其他工具实现整个系统。此阶段应当注意代码质量，遵循编码规范和代码风格。

6.测试验证：测试团队应该对实现的代码进行全面的、系统的测试，以通过测试。在测试过程中，需要重点关注游戏流程、操作手感等用户体验方面。

7.发布部署：部署前端代码到服务器（或云端服务），并确保能够与后端代码进行稳定、无缝的集成。在部署过程结束后，需要进行最后的验收和调整。

总之，一个成功的 World of Zuul 前端开发计划需要团队成员之间的紧密协作和务实监管。各个阶段必须高效且完整地完成，从而为最终的游戏产品带来最佳的用户体验。

项目遵循的主要管理策略和方法：

明确目标，合理制定计划，查询相关资料，按照规定时间完成任务，规范化编写代码

# 4 软件配置计划

1、编码规范：

源代码文件及版本应遵循UTF-8编码，避免出现其他中文及英文字符编码；

1. 命名规范：

文件以及代码中变量命名均遵循规范

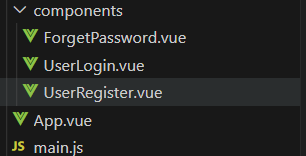


1. 分制管理规范：个人开发过程始终在同一分支下进行，最终统一合并到dev分支中管理

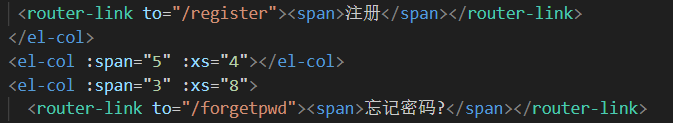
4、提交规范：每次提交都标注好提交内容以及相关说明

# 5 测试计划

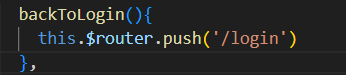
测试过程中主要对三个页面（包括登录，注册，修改密码）的跳转进行了测试



在登录界面使用了router-link 元素进行跳转

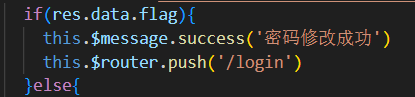


在注册界面使用了this.$router.push 方法进行跳转





在修改密码界面使用了this.$router.push 方法进行跳转



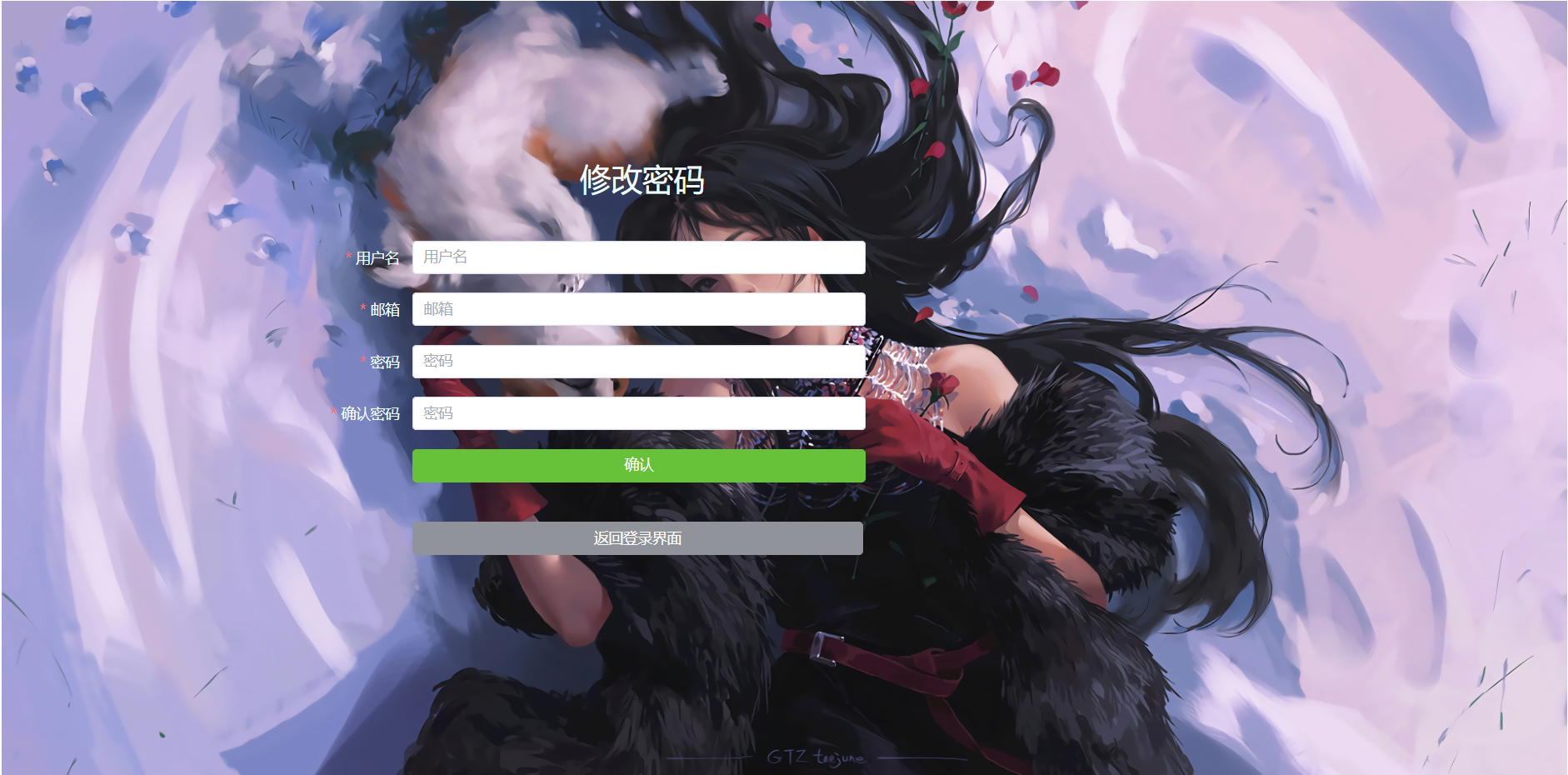
测试结果：



点击登录框中的注册跟忘记密码按钮可以跳转到对应界面

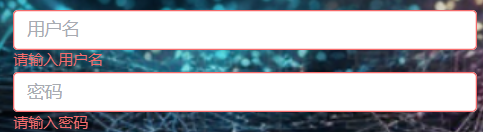
同样，在注册框点击返回登录界面也能顺利返回，在修改密码页面点击返回登录界面也能顺利返回





同时在登录时也进行了登录验证测试：

当输入信息有误时

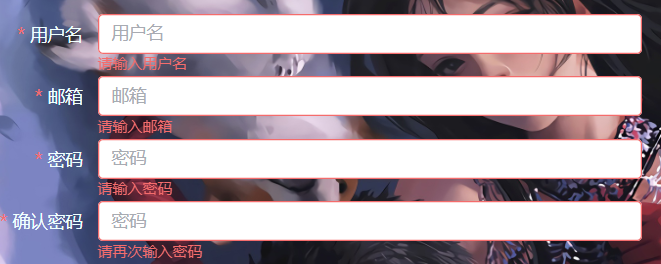


会给出相应提示

在注册和修改密码也进行了类似的测试：



输入信息有误时会给出相应提示



输入信息有误时会给出相应提示

# 6 实施情况

1. Clone对应代码，在本地构建对应仓库即可开始。
2. 登录界面

<1>设计登录页面的UI。登录页面应该能够吸引用户的注意力，并将重点放在登录表单上。视觉设计应该与游戏主题相一致，以增强用户体验。同时，要考虑响应式设计，以确保页面在不同设备上的表现相同。

<2>构建登录表单。登录表单应该包含用户名和密码字段。应该使用调用API获取用户的认证信息，并通过加密过程保持用户数据的机密性。

<3>添加验证逻辑。在提交表单之前，需建立逻辑来验证表单中输入的数据。例如，可以确保用户名和密码格式正确；可以创建一个验证码来防止攻击者破解密码。

<4>处理登录请求。在用户点击“登录”按钮时，前端代码会通过API将表单中输入的数据发送到后端服务器，以验证用户输入的凭据是否与服务器中保存的凭据匹配。如果验证失败，则需要显示相应错误信息。

<5>显示反馈信息。如果验证成功，则用户应被重定向到游戏主界面。如果验证失败，则需要提供适当的反馈信息，以提醒用户输入的凭据非法。

总之，成功实施World of Zuul登录页的关键在于设计一个用户友好、机能齐全的登录表单，并确保前端代码能与服务器端进行稳定的通讯。此外，代码的质量和性能优化将有助于保证正常的游戏体验。

1. 注册跟修改密码页面

<1>设计注册和修改密码页面的UI。页面设计应该吸引用户的注意力，使用户容易理解如何填写信息。应该使用调用API获取用户的认证信息，并通过加密过程保持用户数据的机密性。

<2>构建表单。注册和修改密码表单应该包含必填字段和可选字段。例如，注册表单需要包含用户名、密码和电子邮件地址等必填字段，并包含其他可选字段，如性别、出生日期等。

<3>添加验证逻辑。在表单提交之前，需要建立逻辑来验证表单中输入的数据。例如，可以确保用户名、密码和电子邮件地址格式正确；可以创建一个验证码来验证表单。確保安全性，例如交叉网站脚本攻击（XSS）和请求伪造攻击（CSRF）。

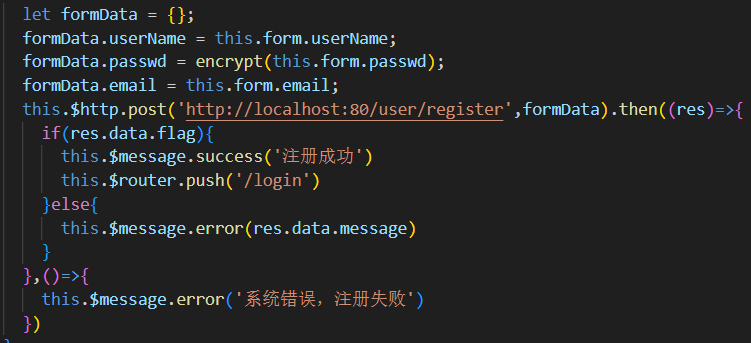
<4>处理注册和修改密码请求。在用户点击“注册”或“修改密码”按钮时，前端代码会通过API将表单中输入的数据发送到后端服务器，以验证用户输入的凭据是否与服务器中保存的凭据匹配。如果验证失败，则需要显示相应错误信息。

<5>显示反馈信息。如果验证成功，则用户应显示成功信息并返回登陆界面。如果验证失败，则需要提供适当的反馈信息，以提醒用户输入的凭据非法。

1. 游戏拓展功能点

在前端页面提供了后端数据库接口，方便记录玩家游戏账号以及对应游戏状态







# 7 实施过程问题记录与分析

1. 页面跳转无响应

更改方法：修改router.js配置

1. 背景图显示有误，页面上方显示一片白色区域

修改方案：更改整体布局，把.el-container属性修改为padding-top

3.游戏主界面（另一人完成） 与登录界面对接，代码合并时出现问题，github网站上显示40+冲突。

解决方案：逐个解决冲突，更改相应代码

# 8 任务总结

需求收集和分析：小组成员根据项目要求，收集和分析用户需求。这个过程包括与项目经理，UI设计师和后端开发人员进行协调，以确保前端面向用户的需求合理。

UI设计：UI设计者负责制定各个页面的美学和功能设计。根据用户需求和游戏主题进行页面UI设计，并制定相应的UI规范和标准。

前端编码实现：前端开发人员根据UI设计师提供的UI及需求分析的要求，使用所选的前端框架和其他工具实现整个系统，并确保代码的稳定性和安全性。

页面交互逻辑实现：交互设计师负责制定网页的互交模式，以优化用户体验。前端开发人员需要根据交互设计师的要求实现页面的交互逻辑功能。

测试和验收：测试工程师对页面进行全面和系统的测试，以确保页面功能满足用户需求，并对市面上同类产品及时进行测试。如果测试成功，项目可以进行交付。

# 9 参考文献

1. **基于Vue的虚拟仿真实验室的可视化设计与搭建 《数码设计（上）》 | 2021 年第 001 期**
2. **基于Vue和Spring Boot的辅导员小助手的设计 《电子技术与软件工程》 | 2021 年第 002 期**
3. **基于Vue构建现代化博物馆可视化应用 电脑知识与技术》 | 2021 年第 014 期**

**《软件工程实践（二）》成绩评定表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓 名** | |  | **学 号** |  | |
| **专业、班级** | |  | | | |
| **成绩评定：** | | | | | |
| 评价内容 | | | 满分 | 实得分 | |
| 得分 | 小计 |
| 实践任务  完成情况 | 软件项目设计、改进与扩充 | | 20 |  |  |
| 个人软件过程与项目管理 | | 15 |  |
| 代码版本管理 | | 25 |  |
| 代码注释与编码规范 | | 25 |  |
| 单元测试 | | 15 |  |
| 实践报告  总评情况 | 学习态度与考勤 | | 10 |  |  |
| 报告格式的规范性 | | 10 |  |
| 报告的逻辑结构与语言表达 | | 15 |  |
| 实践内容的正确性与合理性 | | 60 |  |
| 文献引用及标注 | | 5 |  |
| 总分 | | | 100 |  |  |
| **最终评定成绩（以优、良、中、及格、不及格评定）** | | | | | |

**指导教师签字：**

**年 月 日**